Inf Retrieval J (2016) 19:510–545
DOI 10.1007/s10791-016-9286-2

Evaluation and analysis of term scoring methods for term extraction

Suzan Verberne[1] · **Maya Sappelli**[1,2] · **Djoerd Hiemstra**[3] · **Wessel Kraaij**[1,2]

Received: 15 February 2016 / Accepted: 28 July 2016 / Published online: 10 August 2016

Abstract We evaluate five term scoring methods for automatic term extraction on four different types of text collections: personal document collections, news articles, scientific articles and medical discharge summaries. Each collection has its own use case: author profiling, boolean query term suggestion, personalized query suggestion and patient query expansion. The methods for term scoring that have been proposed in the literature were designed with a specific goal in mind. However, it is as yet unclear how these methods perform on collections with characteristics different than what they were designed for, and which method is the most suitable for a given (new) collection. In a series of experiments, we evaluate, compare and analyse the output of six term scoring methods for the collections at hand. We found that the most important factors in the success of a term scoring method are the size of the collection and the importance of multi-word terms in the domain. Larger collections lead to better terms; all methods are hindered by small collection sizes (below 1000 words). The most flexible method for the extraction of single-word and multi-word terms is pointwise Kullback–Leibler divergence for informativeness and phraseness. Overall, we have shown that extracting relevant terms using unsupervised term scoring methods is possible in diverse use cases, and that the methods are applicable in more contexts than their original design purpose.

Keywords Term extraction · Term scoring · Evaluation · Author profiling · Query expansion · Query suggestion

✉ Suzan Verberne
s.verberne@cs.ru.nl

[1] Radboud University, Nijmegen, The Netherlands

[2] TNO, The Hague, The Netherlands

[3] University of Twente, Enschede, The Netherlands

1 Introduction

Keywords or key *terms* are short phrases that represent the content of a document or a document collection. In some contexts, these terms are formulated by humans, for example by researchers when they submit a manuscript to a journal, or by professionals when they update their online profile. If large collections are involved, or in the context of a system without manual interventions—such as a search system where terms are generated for query expansion—manually selecting terms is not feasible. Automatically identifying terms can then be a good alternative to manually formulating terms. In this paper we adopt the definition of 'terms' by Salton et al. (1976): "appropriate identifiers capable of representing information content". Note that we use the word 'term' to refer to both single-word and multi-word terms. We address the identification of terms as an *extraction task*: The goal of automatic term extraction is to extract and rank the most relevant terms from a document or a document collection. Examples of applications that involve automatic term extraction are: labelling articles in digital libraries with key terms in order to assist browsing by researchers (Gutwin et al. 1999; Witten et al. 1999; Trieschnigg et al. 2009); showing an overview of the contents of a set of retrieved articles in exploratory search (Hofmann et al. 2009); listing topics of expertise on an author profile (Ortega and Aguillo 2014; Verberne et al. 2013); selecting good expansion terms for pseudo-relevance feedback (Cao et al. 2008); extracting potential query terms from clicked documents for personalized query suggestion (Verberne et al. 2014); and finding differences in the language use of two (sub)corpora (Rayson and Garside 2000).

The central methodology needed for term extraction is *term scoring*: each candidate term from the document (collection) is assigned a score that allows for selecting the best—most relevant—terms. The methods for term scoring that have been proposed in the literature were designed with a specific goal in mind, and are used in the literature for a range of diverse applications. It is as yet unclear how these methods compare to each other and how they perform on different types of collections (size, domain, language) than they were designed for. In this paper, we address the following research question:

What factors determine the success of a term scoring method for keyword extraction?

We define *term scoring* as follows: We have a document collection D consisting of one or more documents. Our goal is to generate a list of terms T with for each $t \in T$ a score that indicates how relevant t is for describing D. Each t is a candidate term. t is a sequence of n words: it can be a single-word term or a multi-word term.

In this paper, we evaluate and compare six unsupervised term scoring methods from the literature on four different test collections, each with their own specific use case:

(1) personal scientific document collections; terms are extracted for the purpose of author profiling;
(2) news articles retrieved for Boolean queries; terms are extracted for the purpose of query term suggestion;
(3) scientific articles retrieved for highly specific information needs; terms are extracted for the purpose of personalized query suggestion;
(4) medical discharge summaries; terms are extracted for the purpose of automatically expanding patient queries with medical terms.

A central challenge in our work is the evaluation of the extracted terms. Generally, there are two ways to evaluate terms: intrinsically, by using a (human-defined) ground truth, and extrinsically, using an external application in which the terms are used. This external application then has its own evaluation measure(s). Of the four collections we use for

evaluation, terms that are extracted from collections (1) and (2) are evaluated intrinsically using explicit human relevance assessments; terms extracted from collection (3) are evaluated intrinsically using a partial, human-defined ground truth (terms from the iSearch benchmark data); and terms extracted from collection (4) are evaluated using an extrinsic evaluation measure (ranked retrieval with CLEF benchmark data).

We address the following subquestions:

- What is the influence of the collection size?
- What is the influence of the background collection?
- What is the influence of multi-word phrases?

First, we describe our overall approach in Sect. 2. In Sect. 3 we give an overview of literature on term scoring, and we define and discuss the methods that we implemented. In Sect. 4 we describe the collections that we use for evaluation, followed by a description and discussion of the experimental results in Sect. 5. We conclude the paper with conclusions and recommendations in Sect. 6.

The contributions of this paper are threefold: (1) we do a large-scale evaluation of term scoring methods for term extraction, addressing four different test collections; (2) we not only experimentally evaluate the term scoring methods, but also analyse their scoring functions and show examples of their output; (3) we improve the best performing method by adding a parameter with which the proportion of multi-word terms in the output can be tuned.

2 Our approach

We start by explaining our approach before discussing the term scoring literature and methodology, because understanding the general work flow of our experiments helps understanding the purpose of the term scoring methods we implemented.

Our approach comprises four steps:

1. *Generating candidate terms from the corpus*

In order to generate candidate terms from the document collection D, we first split the collection in sentences, and we extract all word n-grams with $n = \{1, 2, 3\}$ from D that do not cross sentence borders. Then we apply a few filtering rules in order to retain candidate terms: n-grams that do not contain a lowercase letter (`[a-z]`) are skipped, and n-grams that contain a stopword or a 1-letter word are skipped. We do not to apply filtering for part-of-speech patterns because it cannot be known in advance which POS-patterns are relevant for the collection. For example, for some domains we might only be interested in noun phrases as terms, while for another domain verb phrases are important too.[1] Table 1 shows the list of candidate terms extracted for a short example text.

[1] Note that, although the stopword filtering helps in removing many poor terms such as *collection of*, it also results in missing potentially relevant terms such as *learning to rank*. We therefore investigated whether it would be better to keep n-grams with a stopword in the middle. To that end, we extracted all candidate terms from the Wikipedia article "Information Retrieval" (4095 words plain text), thereby only removing the candidate terms that start or end with a stopword. The output contains 279 three-word terms, of which 136 have a stopword in the middle. We went through the list manually and marked for each of the 136 three-wordterms with a stopword as middle word whether or not it is a phrase that should be kept as candidate term. For example, *control a sequence* is a verb phrase, so it is kept, as is the noun phrase *precision and recall*, while the n-grams *backdrop for mechanized*, *graphs which chart* and *importance in automatic* are not kept since they are not syntactic phrases. We found that around half (52 %) of the trigrams with a stopword in the middle are syntactic phrases. This indicates that it might be relevant to keep the n-grams with stopwords in the middle position for a larger coverage of terms; thereby sacrificing precision. This should be investigated in more detail in future work.

2. *Scoring all candidate terms*

We implemented the methods described in Sects. 3.2 and 3.3.

3. *Ranking the terms by their score.*

Depending on the context in which the terms are used, a top-k of the ranked list is returned.

3 Term scoring

Term scoring has been a central topic in information retrieval (IR) since the early years of the field (Salton 1968): In order to find the documents relevant to a user query, both the indexed document and the query are represented as a set of weighted terms that are "appropriate identifiers capable of representing information content" (Salton et al. 1976). The most basic form of term weighting in IR is to give a higher weight to terms that occur more frequently in the document (Luhn 1957). In addition to frequency, term *specificity* is the second cornerstone of term weighting: terms that occur in more documents receive a smaller weight than terms that occur in fewer documents (Sparck Jones 1972). Frequency and specificity were brought together in the famous tf-idf weighting scheme, originally developed for document retrieval (Salton and Buckley 1988) but often used for related tasks such as text categorization (Debole and Sebastiani 2004). Having an index with documents represented by term weights also allows for extracting the most important terms for a document in the index. This principle is applied in pseudo-relevance feedback, where query expansion terms are extracted from the top-ranked documents for the user's query (Xu and Croft 1996; Cao et al. 2008).

The goal of keyword extraction, as we defined it in Sect. 1, is strongly related to this, but more general: terms are extracted from a document *or document collection*, and these terms can be either single words *or sequences of words (multi-word terms)* Each term receives a score that indicates its relevance for the document collection. The input for a term scoring method is an unordered set of candidate terms (see Sect. 2); the output is a score for each candidate term, higher scores indicating more relevant terms. As we will discuss below, frequency and specificity are central components of most term scoring methods, but their operationalizations and implementations differ among methods.

Below, we analyze the characteristics of the methods, in order to provide insight in the strengths of each of the methods before we evaluate them empirically in Sect. 5.

3.1 Term scoring methods

The central component of most term scoring methods is *frequency:* the more often a term occurs in the collection, the more relevant it is for the collection. In the methods we compare, frequency is either

- implemented as raw term count: $count(t, D)$ for a term t in a document collection D,[2] or
- implemented as the maximum likelihood estimate of the probability of occurrence of a term in the collection, i.e. $P(t|D)$ is estimated as the relative term frequency of t in D: $tf(t, D) = \frac{count(t,D)}{|D|}$, in which $|D|$ is the size of D (the total number of words in D).

[2] Note that in the literature, D is often used to denote a single document. We use D to refer to a document collection comprising *one or multiple* documents

Table 1 Candidate terms extracted for a short example text, and the n-grams that were skipped (not saved as candidate terms) for the same text

Example text: Information retrieval is the activity of obtaining information resources relevant to an information need from a collection of information resources. Searches can be based on metadata or on full-text (or other content-based) indexing	
Candidate terms	Skipped n-grams
Information	Is
Retrieval	The
Activity	Of
Obtaining	To
Resources	An
Relevant	From
Need	a
Collection	Activity of
Information retrieval	Relevant to
Obtaining information	Need from
Information resources	Collection of
Resources relevant	Retrieval is
Information need	Resources relevant to
Obtaining information resources	Information need from
Information resources relevant	Can
Searches	Be
Based	On
Metadata	Or
Full-text	Other
Content-based	Searches can
	Based on
	Metadata or

If frequency is used as single measure for relevance, the most relevant terms are generic terms, even if a stopword list is applied. For example, the most frequent non-stopwords in this manuscript are 'terms', 'collection', 'background', 'query' and 'method'. Of these, the first four would be relevant descriptors of this paper, but the last one ('method') is very generic. In addition to that, the most relevant terms will be single-word terms, because the frequency of a term 'x y' in which x and y are single words, can never be higher than the lowest of the two frequencies of x and y. The term scoring methods that we evaluate in this paper therefore extend the frequency criterion with either of two principles: *informativeness* and *phraseness*:

- Informativeness is related to specificity: how much information does a term t provide about D? Most methods for extracting informative terms from a collection use a background collection to determine the informativeness of a term: terms that are much more frequent in D than in a background collection C are the most informative for D. This background collection can be either the collection in which D is included (Hiemstra et al. 2004), or an external collection (Rayson and Garside

2000). An exception is the work by Matsuo and Ishizuka (2004) that exploits the top-k most frequent terms in the document as background model instead.

- Phraseness is a score for how strong (or how 'tight') the combination of words in the multi-word sequence is. Phraseness methods were specifically designed for the extraction of multi-word terms. These methods measure the relevance of a term, using the relative frequencies of these terms and their component unigrams (Tomokiyo and Hurst 2003), or the frequencies of the longer terms in which a multi-word term is embedded (Frantzi et al. 2000).

In the next two subsections we describe the term scoring methods that we evaluate in this paper. All these methods are based on the principles informativeness (Sect. 3.2) and phraseness (Sect. 3.3), all have term frequency as basic component and all are unsupervised, apart from the tuning of a hyperparameter in some methods. In Sect. 3.4 we describe how informativeness and phraseness can be combined in one score. Finally, in Sect. 3.5, we summarize the scoring functions and formulate hypotheses on their strengths.

3.2 Methods for scoring the informativeness of terms

We evaluate four methods that address the informativeness of terms: Parsimonious language models (PLM) by Hiemstra et al. (2004), Kullback–Leibler divergence for informativeness (KLI) by Tomokiyo and Hurst (2003), Frequency Profiling by Rayson and Garside (2000) and the Co-occurrence based method (CB) by Matsuo and Ishizuka (2004). Informativeness methods combine frequency with specificity as measure for the relevance of a term.

3.2.1 Parsimonious language models (PLM)

PLM (Hiemstra et al. 2004) was designed for creating document models in Information Retrieval. In this context, D consists of one document, and it is part of the background collection C. In language models, the background collection is used to smooth the probabilities $P(t|D)$ of terms t in the foreground document D – in order to have no zero probability terms in a document. To that end, linear interpolation smoothing might be used, i.e. a linear combination $\lambda P(t|D) + (1 - \lambda)P(t|C)$, where λ is a smoothing parameter. Parsimonious language models (PLM) re-estimate the probabilities $P(t|D)$ using the following expectation-maximization algorithm.

$$\mathrm{E-step:} \quad \mathrm{e_t} = \mathrm{tf}\ (\mathrm{t,D}) \frac{\lambda \mathrm{P(t|D)}}{(1-\lambda)\mathrm{P(t|C)} + \lambda \mathrm{P(t|D)}} \tag{1}$$

$$\mathrm{M-step:} \quad \mathrm{P(t|D)} = \frac{\mathrm{e_t}}{\sum_{\mathrm{t'}} \mathrm{e_{t'}}} \tag{2}$$

Here, $P(t|D)$ is the probability of the term t in D, $P(t|C)$ is the probability of the term in the background collection and λ is a parameter that determines the strength of the contrast between foreground and background probabilities. In the initialization step, $P(t|D)$ is estimated according to the maximum likelihood estimate in Sect. 3.1. Then the E-step and M-step are repeated for each term t until the estimates $P(t|D)$ converge. The purpose of the iterative EM-algorithm is introducing parsimony: to smooth the document model with the background collection in such a way that a term that is better explained by the background model $P(t|C)$ than by the document model, receives a zero probability for D. This way,

only the most informative terms are kept. In our implementation of PLM, we used three convergence criteria: the relative difference between the probability estimate in two subsequent iterations becoming smaller than 5 %; or $P(t|D)$ becomes smaller than 1 / $|D|$ in which $|D|$ is the number of words in D; or $P(t|D)$ becomes smaller than 0.0001. After convergence, all terms for which $P(t|D) < 0.0001$ are removed from the model.[3]

3.2.2 Kullback–Leibler divergence for informativeness (KLI)

Kullback–Leibler divergence (KLdiv) is a measure from information theory that defines the difference between two probability distributions, in our case the probability distributions of terms in two collections D and C. KLdiv estimates the amount of information that is lost when C is used to approximate D: when the term probabilities for C are used to describe D. *Pointwise* Kullback–Leibler divergence between D and C for a term t is defined as the expected loss of information when the probability of t in C is used to describe the probability of t in D. The terms for which the expected loss of information is the largest are the terms that are the most informative for D (Carpineto et al. 2001; Tomokiyo and Hurst 2003). In the paper by Tomokiyo and Hurst (2003), KLdiv is used for determining term weights according to the two principles of informativeness and phraseness. We will refer to this method as KLIP: Kullback–Leibler divergence for Informativeness and Phraseness. The two components are KLI (for informativeness) and KLP (for phraseness, see Sect. 3.3.2). KLI is defined as:

$$KLI(t) = P(t|D) \log \frac{P(t|D)}{P(t|C)} \tag{3}$$

in which $P(t|D)$ is the probability of the term t in D and $P(t|C)$ is the probability of t in the background collection, both calculated using the maximum likelihood estimate. Since D is not by definition included in C, there may be terms in D that do not occur in C. For these terms, we estimate $P(t|C)$ as 1 / $|C|$, in which $|C|$ is the number of words in the background collection.[4]

3.2.3 Frequency profiling (FP)

This method (Rayson and Garside 2000), designed for contrasting two separate corpora, uses the term frequency lists for both corpora. For each word in the two frequency lists, the log-likelihood (LL) statistic is calculated, based on expected and observed frequencies of a term in both corpora. The expected frequencies of a term in D and C are calculated as follows:

$$E(t, D) = |D| \frac{count(t, D) + count(t, C)}{|D| + |C|} \tag{4}$$

$$E(t, C) = |C| \frac{count(t, D) + count(t, C)}{|D| + |C|} \tag{5}$$

Then, the log-likelihood ratio test (-2LL, as in the original paper) is defined as:

[3] The threshold of 0.0001 was adopted from the original PLM paper (Hiemstra et al. (2004), p.5)

[4] Strictly speaking, $P(t|C)$ is no longer a probability function because $\sum_i P(t_i|C) \neq 1$

$$LL = 2 * (count(t, D) \log \frac{count(t, D)}{E(t, D)} + count(t, C) \log \frac{count(t, C)}{E(t, C)}) \quad (6)$$

The term with the largest LL value is the word with the most significant relative frequency difference between the two corpora. The words that have roughly similar relative frequencies in the two corpora appear lower down the list. The scoring function for FP is similar to the scoring function for KLI. An important difference between FP and KLI is that FP is symmetric and KLI is a-symmetric with respect to the two collections. In other words, FP does not only generate terms that are informative for the foreground collection, but also terms that are informative for the background collection.

3.2.4 Co-occurrence based χ^2 (CB)

In this method (Matsuo and Ishizuka 2004), term relevance for a single document is determined by the distribution of co-occurences of the term with frequent terms in the same document. The rationale of this method is that no background corpus is needed because the set of most frequent terms from the foreground collection serves as background model. χ^2 is then calculated as:

$$\chi^2(t) = \sum_{g \in G} \frac{(count(t, g) - n_t P_g)^2}{n_t P_g} \quad (7)$$

Here, G is the set of 10 most frequent terms in D, $count(t, g)$ is the co-occurrence count (in sentences) of t and $g \in G$, n_t is the total number of co-occurrences of term t and G, and P_g is the expected probability of g:

$$P_g = \frac{n_g^{cooc}}{N} \quad (8)$$

in which n_g^{cooc} is the total term count of terms co-occurring with g in a sentence and N is the total number of terms in the corpus.

Then, the maximum co-occurrence score is subtracted from the total χ^2 in order to discount the score for terms that very frequently co-occur with only one frequent term:

$$\chi^{2'}(t) = \chi^2(t) - \max_{g \in G} \left\{ \frac{(count(t, g) - n_t P_g)^2}{n_t P_g} \right\} \quad (9)$$

3.3 Methods for scoring the phraseness of terms

When using frequency as main criterion for term relevance, multi-word terms are penalized because their frequencies are lower. However, there are many cases where multi-words are highly informative terms. This motivates the design of phraseness methods, which target multi-word terms specifically. We evaluate two methods that address the phraseness of terms: C-Value by Frantzi et al. (2000) and Kullback–Leibler divergence for Phraseness as proposed by Tomokiyo and Hurst (2003).

3.3.1 C-Value

This method (Frantzi et al. 2000) was designed for the recognition of multi-word terms. First, the frequency of each candidate term t (n-gram with $n = \{1, 2, 3\}$ words) in D is determined. This frequency is weighted with the length of t (longer terms get higher weights). Next, a subset T_t is extracted from the set of candidate terms that contains all candidate terms that have t as substring. For example, if t is 'information retrieval' then T_t contains terms such as 'modern information retrieval', 'information retrieval conference' and 'information retrieval journal'. The score for t is discounted with the average frequencies of all $t' \in T_t$. The intuition of the discounting step is that candidate terms that are embedded in frequent longer candidate terms are less informative than terms that are not embedded or only in low-frequent terms. For example, the score for 'language processing' would be heavily discounted because it is embedded in the relatively frequent term 'natural language processing'.

$$\text{C-value}\ (t) = \begin{cases} \log_2 |t| \cdot count(t, D), & \text{if } T_t = \emptyset \\ \log_2 |t| \cdot (count(t, D) - \frac{1}{|T_t|} \sum_{t' \in T_t} count(t', D)), & \text{if } T_t \neq \emptyset \end{cases} \quad (10)$$

where $|t|$ is the length of t (in number of words), $count(t)$ is the number of occurrences of t, T_t is the set of terms that have t as substring and $|T_t|$ is the number of terms in this set. Since $\log_2(1) = 0$, unigrams get a 0-score.

3.3.2 Kullback–Leibler divergence for phraseness (KLP)

As explained in Sect. 3.2, Kullback–Leibler divergence estimates the amount of information that is lost when a *proxy* probability distribution is used to approximate the *target* probability distribution. In the phraseness component of KLIP (KLP), the target probability distribution is the probability distribution for the candidate multi-word term t. The proxy probability distribution is defined as the combined probability distribution of the single words that are contained in t. The terms for which the expected loss of information is the largest are the terms that are the strongest phrases. KLP is defined as:

$$KLP(t) = P(t|D) \log \frac{P(t|D)}{\prod_{i=1}^{n} P(u_i|D)} \quad (11)$$

in which $P(t|D)$ is the probability of t in D and $P(u_i|D)$ is the probability of the ith unigram inside the n-gram t. The intuition is that (a) longer terms get higher weights than shorter terms and (b) relatively frequent multi-word terms that contain at least one low-frequent unigram (e.g. 'ad hoc', 'latent semantic analysis') are the strongest phrases.

3.4 Combining informativeness and phraseness

The only method that has both an informativeness and a phraseness component is KLIP (Tomokiyo and Hurst 2003). In the original paper, KLP is combined with KLI by summing the two scores for one term:

$$\text{KLIP}\ (t) = KLI(t) + KLP(t) \quad (12)$$

We introduce a parameter that allows to combine the informativeness and phraseness components in a weighted sum, adapting Eq. 12: The parameter $\gamma \in [0, 1]$ is the weight of the informativeness score KLI(t) relative to the phraseness score KLP(t):

$$score(t) = \gamma \cdot KLI(t) + (1 - \gamma) \cdot KLP(t) \quad (13)$$

We investigate the effect of γ in Sect. 5.3.

3.5 Hypotheses: strengths of the term scoring methods

Table 2 shows a summary of the term scoring methods described in the previous sections. As introduced in Sect. 1, each method was designed with a specific goal in mind, and they are used in the literature for diverse goals: PLM is generally cited in the context of statistical language modeling for information retrieval (Zhai 2008). CB and KLIP are often used in the context of keyphrase extraction, e.g. in the SemEval tasks (Kim et al. 2013). FP is generally used in corpus linguistics, to study the language use of a particular corpus or genre (e.g. understanding Twitter language (Java et al. 2007)). C-Value is commonly used in the field of Natural Language Processing for the purpose of Information Extraction (e.g. Krauthammer and Nenadic (2004)). Despite these different goals and applications, all methods have common components: they are all based on the pillars frequency and specificity. Therefore, it is to be expected that they are applicable across diverse application domains. For the sake of comparison, we formulate hypotheses about the differences between the methods—both their design purposes and their scoring functions. Our hypotheses focus on the strengths of the methods, related to our three research questions:

1. Collection size: We expect that larger collections will lead to better terms for all methods, because the term frequency criterion is harmed by sparseness. In addition, we expect that PLM is best suited for small collections, because the background collection is used for smoothing the (sparse) probabilities for the foreground collection. Although CB was designed for term extraction from small collections without any background corpus, we do expect it to suffer from sparseness, because the co-occurrence frequencies will be low for small collections. We expect KLIP and C-Value to be best suited for larger collections because of the sparseness of multi-word terms. The same holds for FP, which is similar to KLIP, and was developed for corpus profiling.
2. Background collection: Three methods use a background collection: PLM, FP and KLIP. Of these, we expect PLM to be best suited for term extraction from a foreground collection (or document) that is naturally part of a larger collection, because the background collection is used for smoothing the probabilities for the foreground collection. FP and KLIP are best suited for term extraction from an independent document collection, in comparison to another collection. KLIP is expected to

Table 2 Summary of term scoring methods, with their design purposes

Method	Principle	Designed for modelling a...		Section
CB	I	Single document	Independent of a collection	3.2.4
PLM	I	Single document	As part of a collection	3.2.1
FP	I	Collection	In comparison to another collection	3.2.3
C-Value	P	Collection	Independent of another collection	3.3.1
KLIP	I & P	Collection	In comparison to a background collection	3.2.2 and 3.3.2

In the column 'Principle', I stands for Informativeness and P stands for Phraseness

generate better terms than FP because KLIP's scoring function is a-symmetric: it only generates terms that are informative for the foreground collection.

3. Multi-word terms: We expect C-Value and KLIP to give the best results for collections and use cases where multi-word terms are important. CB, PLM and FP are also capable of extracting multi-words but the scores of multi-words are expected to be lower than the scores of single-words for these methods. On the other hand, C-Value cannot extract single-word terms, which we expect to be a weakness because single-words can also be good terms.

4 Evaluation collections

The subsections below describe the four ollections that we use for evaluation. Each collection is connected to a specific use case. In each subsection, we define the use cases in terms of task, collection and evaluation method. Table 3 at the end of this section shows a summary of the collections.

4.1 Author profiling using a personal scientific document collection

Knowledge workers face enormous amounts of information every day. Not all this information is relevant to the user's current task. Several applications can be envisioned that help knowledge workers to manage (incoming) information: just-in-time recommendation of documents, the automatic filtering of e-mail messages and the personalization of search results. These applications are examples of *personalized information filtering*. For personalized information filtering, a profile of the user is needed that models user-specific terminology. Such a user term profile should serve two purposes (Verberne et al. 2013): (1) it can be used by a filtering tool for estimating the personal relevance of incoming information (documents, e-mails), and (2) it can give the user and his peers insight in his or her profile: which terminology is central in his work? Such a term profile could also be published as an author profile in a digital library or on a personal profile page such as LinkedIn.

Table 3 Summary of the four evaluation collections

Collection	Use case	Evaluation
Personal scientific document collection (English)	Author Profiling using a personal document collection	Intrinsic, using human term judgments
News articles, retrieved with Boolean queries (Dutch)	Query term suggestion for news monitoring (QUINN)	Intrinsic, using human term judgments
Scientific articles, metadata and books (iSearch), retrieved for domain-specific queries (English)	Personalized Query Suggestion	Intrinsic, using ground truth search terms
Discharge summaries (CLEF-eHealth), connected to layman queries (English)	Medical Query Expansion for patient queries	Extrinsic through retrieval task

In the remainder of the article they are referred to by the phrases in boldface

4.1.1 Task

The term scoring algorithm generates terms from a collection of documents and presents them to the user in a ranked list.

4.1.2 Collection and preprocessing

Five knowledge workers provided a collection of documents that are representative for their work (Verberne et al. 2013). The collections consisted of 22 English-language documents on average per user (mainly scientific articles) with an average total of 63,938 words per collecton (standard deviation: 13,583). The document collections were preprocessed by converting each document (from PDF or docx) to plain text and split them in sentences.[5]

4.1.3 Evaluation method

A pool of 150 terms that were scored using three term scoring methods (Hiemstra et al. 2004; Tomokiyo and Hurst 2003; Matsuo and Ishizuka 2004) were judged in alphabetical order by the owner of the document collection. We asked them to indicate which of the terms are relevant for their work (a binary judgment). There was a large deviation in how many terms were judged as relevant by the users (between 24 and 51 %), and on average, 36 % of the generated terms was perceived as relevant (Verberne et al. 2013).[6] Using these relevance judgements, we can calculate average precision (Zhu 2004) for any ranked list of terms:

$$\text{Average precision} = \frac{\sum_{k=1}^{n}(P(k) \times rel(k))}{n_c}, \tag{14}$$

where $P(k)$ is the precision at rank k, n is the total number of terms in the list, n_c is the total number of relevant terms and $rel(k)$ is a function that equals 1 if the term at rank k is a relevant term, and zero if it is not relevant.

4.2 Query term suggestion for news monitoring (QUINN)

LexisNexis Publisher[7] is an online tool for news monitoring. Organizations use the tool to collect news articles relevant to their work. For monitoring the news for a user-defined topic, LexisNexis Publisher takes a Boolean query as input, together with a news collection and a date range. The output is a set of documents from the collection that match the query and the date range. For the users it is important that no relevant news stories are missed. Therefore, the query needs to be adapted when there are changes to the topic. This can happen when new terminology becomes relevant for the topic, there is a new stakeholder or new geographical names are relevant to the topic. Users of news monitoring applications can be supported by providing them with suggestions for query modifications in order to retrieve more relevant news articles. Our intuition is that documents that are relevant but *not* retrieved with the current query have similarities with the documents that *are* retrieved

[5] Sentence splitting was done using the Java text utility java.text.BreakIterator.

[6] Note that it is not possible to calculate inter-rater agreement for this task because only the owner of the document collection can properly judge the relevance of the terms.

[7] http://www.lexisnexis.com/bis-user-information/publisher/

by the current query. Therefore, our approach to query term suggestion is to generate candidate query terms from the set of retrieved documents. This approach is related to pseudo-relevance feedback (Cao et al. 2008), a method for query expansion that assumes that the top-k retrieved documents are relevant, extracting terms from those documents and adding them to the query. There are two key differences with our approach: First, instead of adding terms blindly, we provide the user with suggestions for query adaptation. Second, we have to deal with Boolean queries, without relevance ranking on the retrieved documents. This implies that we do not have a relevance measure for the documents where we extract terms from. This means that the premise of 'pseudo-relevance' may be weak for the set of retrieved documents (Verberne et al. 2015b).

4.2.1 Task

Given a Boolean query, the term scoring algorithm generates terms from the subcollection of documents matching the query and published in the last 30 days, and presents them to the user in a ranked list.

4.2.2 Collection and preprocessing

We collected data in an experiment with 9 experienced Dutch users of LexisNexis Publisher (Verberne et al. 2015b). Together, the users issued 83 searches on LexisNexis' Dutch newspaper collection. The Boolean queries are long: 45 terms on average. The terms can be single words or phrases (multi-word terms), and they are combined with Boolean operators. We used the LexisNexis Publisher API to retrieve documents (news articles) published in the last 30 days. On average, 1031 documents were retrieved per query (ranked by date), with an average length (number of words) of 63.[8] This means that the size of the subcollection from which potential new query terms are extracted for a query is on average $1031 \times 63 = 64,953$ words.

4.2.3 Evaluation method

We collected relevance assessments for the extracted terms in the experiment with 9 users. For the evaluation, we created a pool of terms generated by all term scoring methods. For each method, the top 5 terms are added to the pool. They are ranked by the number of votes they get (the number of methods for which they appear in the top-5 extracted terms). In the experimental interface, the user issues a query in LexisNexis Publisher. The found documents are shown in a result list and a list of query term suggestions (the pool of terms from all methods) is presented. Users were asked to judge the relevance of the returned terms on a 5-point scale (5 meaning 'the term is highly relevant for my information need'), could update the search query (potentially with a suggested term) and retrieve a new result list. We saved the relevance rating for the term, and record the terms that were selected by the user to be added to the query. Then we calculated for each of the term scoring methods two variants of the success rate: (1) the percentages of searches for which the user selected a term from the top-5 and (2) the percentage of searches for which at least one term in the top-5 gets a relevance rating $> = 4$.

[8] The short document length is caused by the API allowing us to extract only the summary of the news article, not the full text.

4.3 Personalized query suggestion

The previous task (QUINN) was query suggestion for longitudinal Boolean queries that are used for news monitoring. In the context of web search, query suggestion is a functionality of a search engine that suggests the user a list of queries to proceed the search session with. If the query suggestion algorithm works well, it reduces the cognitive load of users and makes them more efficient in their search for information (Azzopardi et al. 2013). For web search, query logs are a good source for query suggestion (Huang et al. 2003). However, for search tasks addressing highly specialized topics, where there are no relevant queries from other users available, the alternative is to fall back to the user's own data (Shen et al. 2005). In personalized interactive search, the initial query is formulated by the user; query suggestion can assist the user in entering effective follow-up queries (Verberne et al. 2014). The documents that the user clicks on are a good source for query terms that can improve the user's query because they are likely to be related to the user's information need. Thus, term extraction in this task is directed at generating potential query terms from relevant documents. For each topic, a subcollection of relevant documents is created using the relevance judgments provided with the data, as source for term extraction.

4.3.1 Task

The term scoring algorithm generates candidate query terms from the subcollection of relevant documents and presents these terms (extensions or adaptations of the previous query) to the user in a ranked list.

4.3.2 Collection and preprocessing

The iSearch collection of academic information seeking behavior (Lykke et al. 2010) consists of 65 English-language natural search tasks (topics) from 23 researchers and students from university physics departments. The topic owners filled in a form with five fields, among which an explicit description of their information need, and a list of search terms that they would use to express this information need. A collection of 18K book records, 144K full text articles and 291K metadata records from the physics field is distributed together with the topics. Relevance judgments are provided for 200 documents per topic. Since we do not have user interactions (clicks or simulated clicks) available in the current study, we use the subset of relevant documents for a given topic as subcollection. The average number of relevant documents for a topic is 42. For the documents in the subcollection, the fields 'title' and 'description' are included in the case of metadata and book records and the first 200 words in the case of articles in PDF (for which no metadata is available). The collection size per topic is 2250 words on average.

4.3.3 Evaluation method

For this task we have a small but exact set of reference terms: the list of search terms provided by the topic owners in the iSearch data. We consider these terms to be the ground truth for query formulation. We evaluate the list of ranked terms from the subcollections using Average Precision (see Eq. 14), with the ground truth terms as reference for relevance. The set of reference search terms is small, and likely to be different than terms generated from retrieved documents: the human-formulated search terms are long and

highly precise (e.g. 'Induced-charged electro-osmosis', 'Coupled photonic crystal cavity lasers'). Therefore we expect a relatively low Average Precision for this task. Since we are interested in the relative performance of the methods we evaluate, this is not necessarily problematic: the higher the ranks of the reference terms in the returned term list, the better the term scoring method.

4.4 Medical query expansion for patient queries

This collection was created for CLEF eHealth 2014, task 3a.[9] The motivation for the task is as follows: Often, a patient starts searching the internet for medical information about his illness after he has learned from his physician what his diagnosis is. The goal is to retrieve the most relevant medical information for a patient's query. The physician's information about the patient has been registered in the patient's *discharge summary*. The patient uses 'layman' query terms, while the discharge summary contains an expert description of the diagnosis (Goeuriot et al. 2014; Kelly et al. 2014). Since the discharge summary is on the same topic as the query, but uses a different vocabulary, it might contain useful query terms that can be used to retrieve additional relevant medical information (Verberne 2014). Thus, the purpose of term extraction for this task is to expand the original query with key terms extracted from the discharge summary. In order to find a successful strategy for query expansion using extracted terms, we turned to the methods applied by teams participating in the task. The most successful teams were Choi and Choi (2014), Oh and Jung (2014) and Shen et al. (2014).

Oh and Jung (2014) implement and evaluate five steps of document re-ranking. The second step is query expansion with terms from the discharge summary, which they find to have a positive effect on the retrieval effectiveness. Unfortunately, they do not specify how many terms from the discharge summary they add to the query, nor the weight that they assign to the expansion terms. Choi and Choi (2014) do not use the discharge summary for extracting terms but expand the user query with terms from the UMLS, followed by a learning-to-rank approach using document features. Shen et al. (2014) also use UMLS based lexical query expansion. They compare multiple operators in the Indri query language to combine terms: `#1()` (treating the string between brackets as a literal phrase) `#combine()` (treating the string between brackets as a bag of words) and `#uwN()` (all words between brackets must appear within window of length N in any order).[10] They find that `#uwN` is the most powerful operator. In Sect. 5.1.2, we describe our strategy for query expansion with terms from the discharge summary, based on these findings.

4.4.1 Task

The term scoring algorithm generates terms from the discharge summary to be added to the query.

4.4.2 Collection and preprocessing

As evaluation set we use the training and test collections from CLEF eHealth task 3a (Kelly et al. 2014): the CLEF document collection and five train + 50 test topics (layman's information needs in English) with a discharge summary for each topic. We used

[9] See http://clefehealth2014.dcu.ie/task-3

[10] See http://www.lemurproject.org/lemur/IndriQueryLanguage.php

the Indri API to index the CLEF collection and set up a query interface to the index. A corpus of 299 English-language discharge summaries was distributed for CLEF-eHealth (Kelly et al. 2014). We cleaned the discharge summaries from all variables of the form `[** ...**]` (e.g. `[**MD Number 2860**]`),which were added by the track organizers for the purpose of data anonymization. A topic in the CLEF-eHealth task consists of five descriptive fields: title, description, profile and narrative. We use the title field, or the title together with the description as query. For query construction, all characters that are not alphanumeric, not a hyphen or whitespace are removed from the query and all letters are lowercased. The words in the query are concatenated into one string and combined using the `combine` function in the Indri query language. The result is the baseline query for the topic that will be expanded with terms from the discharge summary.

4.4.3 Evaluation method

We do not have a list of relevant terms from the discharge summary. We therefore evaluate the extracted terms extrinsically, by using them as additional query terms for retrieving documents from the CLEF collection: an increasing number of top-ranked terms (0,2,5,10,20) are added to the baseline query. With the resulting expanded query, 100 documents are retrieved from the CLEF collection and ranked using the Indri LM with Dirichlet smoothing. We evaluate the retrieval effectiveness in terms of nDCG, one of the most used evaluation measures for ranked retrieval (Järvelin and Kekäläinen 2002).

5 Experiments with term scoring methods

In the next three subsections, we address the three research questions from Sect. 1 with a series of experiments:

1. What is the influence of the collection size? (Sect. 5.1)
   - The influence of collection size on the effectiveness of term scoring (5.1.1)
   - Comparing methods for small data collections (5.1.2)
2. What is the influence of the background collection? (Sect. 5.2)
   - Comparing methods with different background corpora in the Personalized Query Suggestion collection (5.2.1)
   - Comparing methods with different background corpora in the QUINN collection (5.2.2)
3. What is the influence of multi-word phrases? (Sect. 5.3)

In each subsection, we address two of the four evaluation collections. Table 4 shows an overview. Each subsection is concluded with a discussion of the experimental results in the light of the hypotheses in Sect. 3.5.

5.1 What is the influence of the collection size?

Table 5 shows the sizes of the four document collections. It shows that the Author Profiling and QUINN collections are large, and that the other two are relatively small in terms of number of words. QUINN has a large number of documents but since we only have

Table 4 Overview of experiments per research question

Section	RQ	Evaluation 1	Evaluation 2
5.1	Collection size	Author profiling	Medical query expansion
5.2	Background corpus	Personalized Query Suggestion	QUINN
5.3	Multi-word terms	Author profiling	Personalized Query Suggestion

access to the abstracts of news articles, the document length is small (63 words on average). In Personalized Query Suggestion, the number of documents is reasonable, but the documents are also relatively short, since they consist of metadata or the first 200 words of a pdf. The collections in Medical Query Expansion are the smallest, with only 1 document of 609 words on average per topic.

We address two collections in this section: the Author Profiling collections, where we evaluate term scoring for increasing word counts, and discharge summaries for Medical Query Expansion, where we investigate how different methods perform on collections with a small number of words.

5.1.1 The influence of collection size on the effectiveness of term scoring

We investigate the effect of the collection size by manipulating the Author Profiling collections as follows: we split all documents from the collection in paragraphs, randomize the order of the paragraphs, and then create subcorpora with increasingly more paragraphs from the collection, up to {100, 500, 1000, 5000, 10,000, 20,000, 30,000, 40,000, 50,000} words. We then evaluate term extraction for each subcorpus. The reason that we increase the size of the corpus by paragraph and not by document, is that documents are relatively long and covering one topic each, as a result of which the presence or absence of a complete document will strongly influence the presence or absence of topics in the list of extracted terms, especially in the smaller collections. The randomized sampling of paragraphs ensures a smoother curve. Because of the randomization component, we run each experiment five times and report averages over these five runs.

We evaluate all five term scoring functions for the increasing collection size.[11] For PLM, we set $\lambda = 0.1$, which was suggested as optimal in the original paper (Hiemstra et al. 2004). PLM, FP and KLIP (KLI) require a background collection. We used a corpus of generic English for this, the Corpus of Contemporary American English (COCA) (-Davies 2009), which contains 450 Million words. The owners of this corpus provide a word frequency list and n-gram frequency lists that are free to download.[12]

Figure 1 shows mean average precision scores over the users in the Author Profiling data for increasing collection sizes. For CB, we evaluated both $|G| = 10$ and $|G| = 100$ for the reference set of top-frequent terms G and they give almost the same results. Apparently, the distribution of co-occurrence frequencies does not change much when we use a larger reference set of top-frequent terms in the collection. Therefore, we only show the results for $|G| = 10$ here. Of the informativeness methods, PLM, KLI and FP give better results than CB. The results also show that KLI and FP reach their maximum effectiveness

[11] When running C-Value, we remove n-grams with a frequency lower than 5 from the candidate termset to reduce the processing time of finding all terms that have t as substring for each t in the termset.

[12] http://www.wordfrequency.info/

Table 5 Sizes of the four document collections

Collection	No. of docs		No. of words	
Author Profiling	22	Docs (avg per user)	63,938	(avg per user)
QUINN	1031	Docs (avg per query)	64,953	(avg per query)
Personalized Query Suggestion	42	Rel docs (avg per topic)	2250	(avg per topic)
Medical query expansion	1	Discharge summary	609	(avg per topic)

at a collection size of 20,000 words, and do not improve anymore with increasing collection sizes. PLM and CB reach their maximum earlier: PLM does not improve after 10,000 words and CB's effectiveness improves only slightly after 1000 words, but not anymore after 5000 words. This is not surprising giving the original purpose of the methods: PLM and CB were designed for term extraction from a single document.

The phraseness methods behave interestingly. We see that both KLP and C-Value perform better than any of the informativeness methods for collections larger than 20,000 words. There are two reasons for that: First, multi-word terms are important for the scientific domain and judged as better terms by human assessors and second, multi-word terms are less sparse in larger collections.

The graph also shows that KLP performs better than C-Value. This is an interesting finding because the two methods use different criteria for selecting terms: both favor longer terms over shorter terms, but in C-Value, the score for a term is discounted if the term is nested in frequent longer terms; in KLP, the frequency of the term as a whole is compared to the frequencies of the unigrams that it contains. Thus, KLP prefers frequent multi-word terms consisting of lower-frequent unigrams, while C-Value prefers terms that are not nested in longer terms. Table 6 shows example output for KLP and C-Value to illustrate this difference. For completeness, the example output for the informativeness methods is also added to the table.

The lists for KLIP and C-value are similar, showing largely the same terms, although their ranks are different. Terms that are selected by KLP and not by C-Value are 'new york' and 'entity ranking topics'. Terms that are selected by C-Value and not by KLP are 'category information' and 'target categories'. 'new york' is probably the most clear example of the difference between the methods: in this corpus, the term 'new york' is almost as frequent as the unigram 'york'. In other words, 'york' almost only occurs together with 'new', which makes 'new york' a very tight n-gram, and therefore a strong phrase for the KLP criterion. For C-Value however, the phrase is not very strong because it is nested in a number of frequent longer phrases such as 'new york university' and 'new york ny'.

5.1.2 Comparing methods for small data collections

As shown in Table 5, the Medical Query Expansion data collection is small (1 document of 609 words on average per topic). Therefore, we use this collection to evaluate the performance of the term scoring methods for small data collections. Medical Query Expansion is a use case with an extrinsic evaluation measure: nDCG for the set of retrieved documents (see Sect. 4.4). In order to evaluate the term scoring methods, we extract terms from the discharge summary belonging to the topic and add an increasing number of top-ranked terms (0,2,5,10,20) to the query. Table 7 shows an example query with expansion terms.

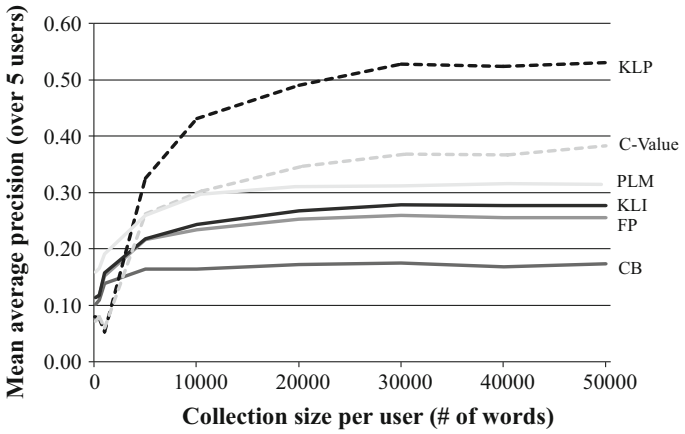

Fig. 1 The effect of collection size on the performance for five different term scoring methods on the Author Profiling collections. The solid lines represent the informativeness methods; the dashed lines represent the phraseness methods. KLI is KLIP with $\gamma = 1$ (informativeness only) while KLP is KLIP with $\gamma = 0$ (phraseness only). Each point in the graph is an average over 5 runs because of the randomized data selection

Table 6 Example output of each of the term scoring methods for one of the Author Profiling collections: the top-10 terms of the expert profile generated from the collection of scientific articles authored by one person, who has obtained a PhD in Information Retrieval

Phraseness methods		Informativeness methods		
KLP	C-Value	PLM	KLI	FP
Entity ranking	Entity ranking	Category	Pages	Pages
Ad hoc	Anchor text	Categories	Categories	Categories
Anchor text	Ad hoc	Query	Query	Query
Test persons	Test persons	Entity	Results	Results
et al	Relevance feedback	Pages	Using	Using
Word clouds	Language model	Using	Retrieval	Retrieval
Relevance feedback	Word clouds	Results	Documents	Documents
New york	et al	Retrieval	Topical	Entity
Language model	Category information	Documents	Wikipedia	Category
Entity ranking topics	Target categories	Information	Topics	Topical

In a short CV, she describes her research topics as "entity ranking, searching in Wikipedia, and generating word/tag clouds"

We experiment on the training set provided by CLEF (5 topics) with the following settings for query expansion:

(a) the length of the original query: using only the words from the title of the topic or words from the title and the description of the topic;
(b) the operator for multi-word terms: #1, #2 or #uw10;[13]

[13] See http://www.lemurproject.org/lemur/IndriQueryLanguage.php for a definition of the operators.

Table 7 Example query from the CLEF eHealth data for the Medical Query Expansion collection with the top-5 terms extracted from the discharge summary using five different term scoring methods

Title from CLEF topic:	`<title>Esophageal perforation and risk</title>`
Indri query (topic title):	`# combine(esophageal perforation and risk)`
Top-5 terms from discharge summary added to query:	
PLM	mg, patient, day, hospital, tube
KLIP	mg, hospital day, ampicillin gentamicin, three times, ampicillin
CB	mg, day, patient, patients, hospital
FP	mg, ampicillin, hospital day, avonex, baclofen
C-Value	hospital day, three times, ampicillin gentamicin, location un, advanced multiple sclerosis
Example of expanded Indri query	`#combine(esophageal perforation and risk #weight( 0.024382201790445927 mg 0.01744960633704929 #2(hospital day) 0.016052177097263427 #2(ampicillin gentamicin) 0.013107586537605164 #2(three times) 0.011385981676144982 ampicillin ))`

(c) the weights for the expansion terms: uniform (each term gets as weight 1 / T, where T is the number of expansion terms) or the term score that each term received from the term scoring algorithm.

For PLM, we optimize the parameter λ on the training set, investigating values ranging from 0.0001 to 1.0, of which 0.01 turned out to be optimal. For KLIP, we set $\gamma = 0.5$. We found that title-only gave better results than title+description; that the operator #2 was slightly better than the other two, and that term scores as weights were a bit better than uniform weights. Below, we show the results obtained on both the training set (5 topics) and the test set (50 topics) for these settings. The bottom row of Table 7 shows an example of an expanded Indri query.

The results are in Fig. 2. Surprisingly, we seem to obtain positive results on the training set that are not replicated on the larger test set. The mean nDCG for the test queries without expansion terms is very close to the mean nDCG for the train queries, but adding terms from the discharge summary does not give the seemingly positive effect that it has on the training set. Since the training set is small (only 5 topics), we suspect that the different behaviors between train and test set are due to individual differences between topics. The graphs in Fig. 2 represent averages over all topics; the standard deviations are relatively large: between 0.20 and 0.23 for each point in the graphs. There are topics for which the expanded terms have a positive effect, and there are topics for which they have a negative effect, and there are topics for which they have no effect. A closer look at the top-10 extracted terms for each of the termscoring functions shows that the 20 most occurring terms are the following:

mg	tablet	right	blood pressure
sig one	one	mg tablet sig	admission date
mg po	sex	tablets	tablet sig
patient	sig	po	day
mg tablet	discharge	one tablet	tablet sig one

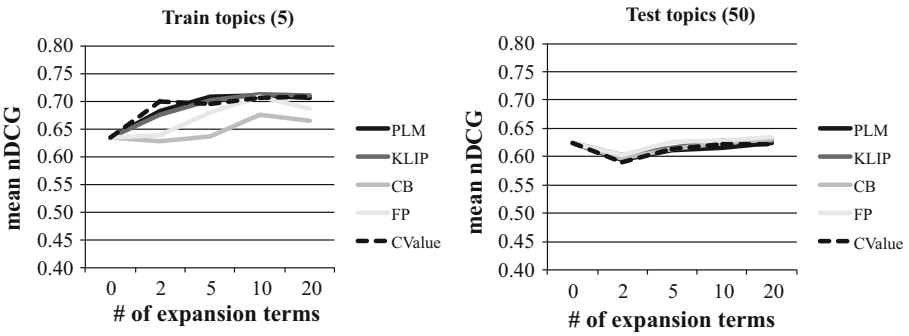

Fig. 2 The effect of query expansion with terms extracted from discharge summary (the Medical Query Expansion collection) using five different term scoring methods, in terms of nDCG

These are all generic terms in the medical domain. If we look at the frequencies for the top-term 'mg', we see that it occurs dozens (> 30) of times in each of the discharge summaries in our set, and although it is also frequent in the background collection of discharge summaries (1,266 occurrences on a total term count of 194,406), its high frequency in the foreground collection still make it a good term according to the term scoring functions, which all have term frequency as their most important component. More specific terms, such as medicine names (e.g. glipizide, risperidone) occur lower in the term lists; their absolute frequencies are much lower: below 5. It seems that all methods are hampered by the small collection size (609 words on average per discharge summary), combined with the semi-structured nature of the texts in which there is a lot of repetition of technical phrases such as 'mg po' and 'sig one'.

5.1.3 Discussion: What is the influence of the collection size?

In Sect. 5.1.1 we studied the effect of collection size for a use case with a human-defined ground truth: Author Profiling. We found that larger collections lead to better terms. PLM gives the best results for collections smaller than 5,000 words, while both KLP and C-Value perform better than any of the informativeness methods for collections larger than 20,000 words. KLI and FP reach their maximum effectiveness at a collection size of 20,000 words; PLM at 10,000 words and CB at 5,000 words. The poorest performing method is CB. This is the only informativeness method that does not exploits a background collection for calculating the informativeness of terms, but instead uses the set of frequent terms in the foreground collection as a proxy for a background collection. A method that does not require a background collection could be appealing, because it eliminates the choice for a background collection, but apparently, the set of frequent terms from the foreground itself is a weak background model. This confirms our hypothesis:

> **Hypothesis:** We expect that larger collections will lead to better terms for all methods, because the term frequency criterion is harmed by sparseness. In addition, we expect that PLM is best suited for small collections, because the background collection is used for smoothing the (sparse) probabilities for the foreground collection. Although CB was designed for term extraction from small collections without any background corpus, we do expect it to suffer from sparseness, because the co-occurrence frequencies will be low for small collections. We expect KLIP and

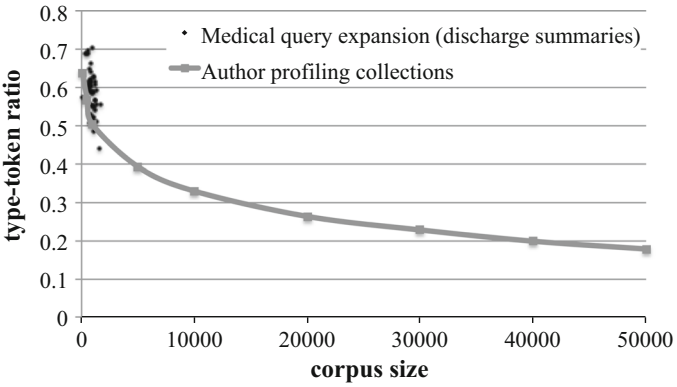

Fig. 3 The type-token ratio as a function of corpus size, for the author profiling and the Medical Query Expansion collections. Each point in the author profiling graph is an average over 5 runs like in Fig. 1. Each dot in the Medical Query Expansion graph represents one discharge summary (the foreground collection for one topic)

C-Value to be best suited for larger collections because of the sparseness of multi-word terms. The same holds for FP, which is similar to KLIP, and was developed for corpus profiling.

In Sect. 5.1.2, we found that all methods are hindered by small collection sizes (a few hundred words): the absolute frequencies of specific terms are low and 1 or 2 additional occurrences of a term makes a large relative difference.

In order to provide more insight in the effect of corpus size on term extraction performance, we investigated the type-token ratios for the author profiling and the Medical Query Expansion collections. Type-token ratio (TTR) is a measure of lexical variety: it gives the ratio between the number of unique words (types) and the total number of words (tokens) in a corpus. It has been reported before that TTR is related to corpus size: the larger the corpus, the lower the TTR (?). A high type-token ratio indicates that many terms only occur once, as a result of which the frequency criterion bears little relevance. Since the frequency criterion is central to all term scoring methods, we would expect the methods to perform poorly on collections with a high TTR. Figure 3 shows the TTR als function of the corpus size, for both collections. The TTR graphs confirm the relation between TTR and corpus size. It shows that the Medical Query Expansion collections have a high type-token ratio, 0.59 on average, with an average corpus size of 609. The TTR for the author profiling collections at this corpus size is similar: the gray line is very close to the black dots. In Fig. 1, we see that for this corpus size, all term scoring methods perform poorly relative to their performance with the maximum corpus size: between 0.05 and 0.20, while they reach between 0.18 and 0.53 at their maximum.

This analysis confirms our finding that the term scoring methods all perform poorly on small corpus sizes. We speculate that this is caused by the prominence of the frequency criterion in all methods: For small collections term frequency is a weak variable: most terms occur only once or a few times.

5.2 What is the influence of the background collection?

The choice of the background collection depends on the language and domain of the foreground collection, and on the purpose of the term extraction. In this section, we

evaluate the effect of the background corpus in three informativeness methods (PLM, KLIP (KLI) and FP), for two collections: Personalized Query Suggestion, where we compare a generic and a domain-specific background corpus, and QUINN, where we compare the use of an external background corpus (a Dutch news corpus) and the use of a topic-specific collection: an older subcollection of documents for the same query.

5.2.1 *Comparing methods with different background corpora in the personalized query suggestion collection*

We first investigate the effect of the parameter λ in the PLM method. λ defines the weight of the background collection in smoothing the term probabilities for the foreground collection. We extract terms from the subcollection of relevant documents using PLM, with two different background collections: the iSearch collection (which would be the 'natural', domain-specific background corpus for this collection) and COCA (which is an external corpus, with general language).

We use the topics 001–031 from the iSearch data to optimize the parameter λ and we investigate values of λ ranging from 0.0001 to 1.0. The results are in Fig. 4. Note that $\lambda = 1.0$ is the setting in which the background corpus frequencies are not used at all and the algorithm does not change the initial values of $P(t|D)$. The plot shows that (a) Mean Average Precision is low for this collection. This is because the ground truth is very strictly defined; we did not collect relevance assessments for all returned terms; (b) iSearch as background corpus seems to give better results than COCA, but this difference is not significant (for the λ-value with the largest difference, $\lambda = 0.01$, a paired t test on the AP-scores for individual topics gives $p = 0.263$ for the difference between COCA and iSearch); (c) the effect of λ is almost negligible for COCA, but shows a peak for iSearch at 0.01.

We investigated the output of the EM-algorithm over the iterations in order to find out why λ has little effect for these data. We see that for most topics, only two or three iterations are needed for the estimated probabilities to converge. We speculate that since the most informative terms converge very fast, the contrast of their frequencies between the foreground and the background corpus is apparently sufficiently large to receive a high probability, independent of the weight of the background corpus.

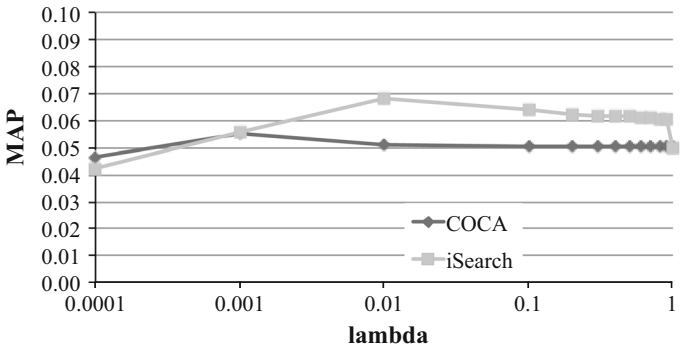

Fig. 4 The effect of the parameter λ in the PLM method, for the Personalized Query Suggestion collection, with two different background corpora: the collection of which the foreground collection is a subset (iSearch) and an external collection with generic English (COCA). The x-axis uses a log-scale

Table 8 The effect of the background corpus in three different informativeness methods, for the Personalized Query Suggestion collection, in terms of Mean Average Precision

	COCA (SD)	iSearch (SD)	*P* value for the difference
PLM ($\lambda = 0.01$)	0.028 (0.050)	0.042 (0.087)	0.152
FP	0.025 (0.043)	0.040 (0.072)	0.042
KLIP ($\gamma = 1.0$)	0.026 (0.047)	0.038 (0.069)	0.076

P values are calculated using a paired *t* test with the scores paired per topic

In the remainder of this section, we use $\lambda = 0.01$ for PLM. For KLIP, we set $\gamma = 1.0$ because we evaluate the informativeness component and not use the phraseness component. We use the topics 032–066 from the iSearch data to compare the methods. The results are in Table 8.

Table 8 shows that the domain-specific iSearch corpus gives better results than the generic COCA for all three methods. For FP, this difference is significant at the 0.05-level. The differences between the three methods PLM, FP and KLIP are not significant on the 0.05 level: a paired *t* test for the largest difference (between KLIP and PLM with iSearch as background collection) gives $p = 0.111$. Table 9 illustrates the output for the FP method with the two different background corpora. Many terms overlap, although their ranking is different.

In Sect. 5.2.3 we come back to these results and provide some more insight on the effect of the background collection.

5.2.2 *Comparing methods with different background corpora in the QUINN collection*

For the QUINN collection, we compare two different background corpora for extracting potential query terms from news articles of the last 30 days for a given query:

(a) an older result set for the same query: all news articles matching the query that were published between 60 and 30 days ago;
(b) a generic news collection. Since the QUINN collection is Dutch, we use the newspaper section from the SoNaR-corpus (Oostdijk et al. 2008), 50 Million words in total, for this purpose.[14]

Of these two corpora, (a) is topic-related and thereby highly domain-specific, even more than the iSearch corpus was for Personalized Query Suggestion in academic search (see the previous section), and (b) is more generic but from the same genre as the foreground collection (Dutch newspaper texts).

We use both background corpora for extracting terms with PLM, FP and KLIP ($\gamma = 0.5$) and evaluate the quality of the extracted terms using two user-based evaluation measures: the percentage of searches with a term from top-5 selected by the user, and the percentage of searches with at least 1 relevant term (a relevance rating $> = 4$ on a 5-point scale) in top-5. The results are in Figs. 5 and 6.

The figures show consistently better results for the generic newspaper background corpus than for the topic-related background corpus. A McNemar's test for paired binary

[14] Corpus available at http://tst-centrale.org/producten/corpora/sonar-corpus/6-85

Table 9 Example output of FP with iSearch and COCA as background corpus for the Personalized Query Suggestion collection: the top-10 terms extracted from the relevant documents in the iSearch collection for one topic (045), "Models of emerging magnetic flux tubes"

FP with iSearch	FP with COCA
Magnetic	Magnetic
Solar	Flux
Coronal	Fields
Flux	Simulations
Magnetic flux	Solar
Corona	Coronal
Convection	Corona
Tube	Heating
Magnetic fields	Convection
Tubes	Magnetic flux

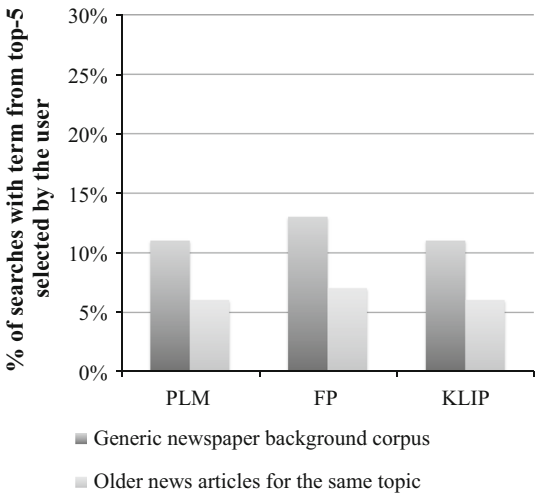

Fig. 5 The quality of the suggested query terms in QUINN, using three different methods and two different background corpora, in terms of the percentage of searches with a term from top-5 selected by the user

samples[15] shows that the difference between the two corpora is significant on the 0.01 level for PLM ($p = 0.0036$) and on the 0.05 level for FP ($p = 0.037$) and KLIP ($p = 0.034$). It is surprising that the generic background corpus gives better results than the domain-specific corpus, considering the results in the previous subsection, where the domain specific iSearch corpus seemed to be give better results than the generic COCA. We had a detailed look at the terms generated using either of the two background corpora. Two example queries with their term suggestions are shown in Table 10.

In the example on Biodiversity, the terms generated with two background corpora show quite some overlap, but in the example on ICT policy, the two term lists are completely different. In both cases, the terms generated with the topic-related background corpus are more specific than the terms generated with the generic background corpus. In other words, the comparison between the news from the last 30 days to a generic newspaper corpus leads to terms that are relevant for the topic in general, while the comparison between the

[15] $N = 83$; each query is labeled '1' if the suggestion list contains at least one relevant term and '0' if there are no relevant terms suggested

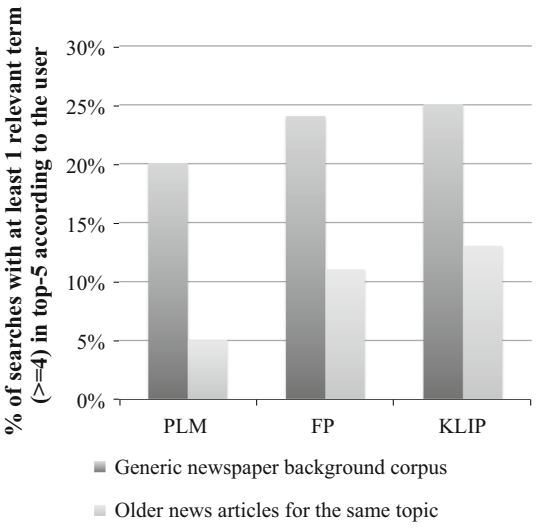

Fig. 6 The quality of the suggested query terms in QUINN, using three different methods and two different background corpora, in terms of the percentage of searches with at least 1 relevant term (a relevance rating $> = 4$ on a 5-point scale) in top-5

news from the last 30 days and the news on the same topic from 60-30 days ago leads to terms that are very specific for the most recent developments on the topic. Hence, the second example topic contains a few names of places (Westrozebeke, Moorslede) that were in the news during the last 30 days. This leads us to the conclusion that a domain-specific background corpus is good, but this domain should not be too narrow (such as a corpus covering one news topic).

5.2.3 Discussion: what is the influence of the background collection?

Since the term scoring methods were designed for different purposes, the choice of background corpus and the term scoring method are expected to be interdependent. Specifically, PLM was designed for modelling a single document in the context of a larger collection, while KLIP and FP were designed for contrasting two collections. Hence our hypothesis:

> **Hypothesis:** Three methods use a background collection: PLM, FP and KLIP. Of these, we expect PLM to be best suited for term extraction from a foreground collection (or document) that is naturally part of a larger collection, because the background collection is used for smoothing the probabilities for the foreground collection. FP and KLIP are best suited for term extraction from an independent document collection, in comparison to another collection. KLIP is expected to generate better terms than FP because KLIP's scoring function is a-symmetric: it only generates terms that are informative for the foreground collection.

With term extraction for query suggestion in the scientific domain (the Personalized Query Suggestion collection, Sect. 5.2.1), we had relatively small collections—2250 words on average per topic—that are part of the background collection. For this type of collections we would expect that PLM would outperform FP and KLIP. The results that we got in terms of Mean Average Precision (Table 8) seem to indicate that PLM indeed is a bit better than the other methods, but these differences are not significant. This is probably due to the strictly defined baseline (a small set of human-formulated query terms). Throughout

Table 10 Generated terms for two example topics using PLM with two different background corpora

Topic: Biodiversiteit 'Biodiversity'	
Query: (Biodiversiteit AND (natuur! or rode lijst! or planten or dieren or vogels or vissen or zee! or zeeen or oceaan or oceanen or exoten or uitheemse flora or uitheemse fauna or inheemse planten or inheemse dieren or inheemse flora or inheemse fauna or duurzaamheid or soorten!)) OR otter OR gierzwaluw OR kiekendief OR trekvogel AND NOT vogelgriep OR ...)	
Generic newspaper background corpus	Topic-related background corpus
natuur 'nature'	vogelteldag 'bird count day'
hectare 'hectare'	spreeuw 'starling'
vogelteldag 'bird count day'	getelde vogel 'counted bird'
trekvogels 'migrating birds'	vaakst 'most often'
spreeuw 'starling'	getelde 'counted'
Topic: ICT beleid 'ICT policy'	
Query: (sms w/4 (gedragscod! OR meldpun!)) OR (overstap! w/p (telefo! OR internet!)) OR telemarket! OR ((telecomwet! OR regule! OR wet OR wetten OR wetg!) AND (internet! OR cookie!)) OR ((veilen OR geveild OR veiling!) w/p frequenti!) OR frequentieveil! OR (marktrapportag! w/s ele?tron! communic!) OR digitale agenda! OR overheidsdata OR ict office OR ecp epn OR logius OR digipoort OR (duurza! w/s ict) OR (energie! w/s ict) OR (declaration w/2 amsterdam) OR (verklaring w/2 amsterdam) OR WCIT OR (world congress w/s allcaps(IT)) OR (SBR AND NOT bouw) OR standard business reporting OR (mobiel w/2 betalen) OR (betalen w/3 (telefoon OR mobiel OR gsm)) OR sggv OR slim geregeld goed verbonden OR (eod AND NOT explosieven!) OR ele?tron! ondernem! OR ele?tron! zaken! OR (Besluit Universele Dienstverlening w/s Eindgebruikersbelangen) OR apps for amsterdam OR apps for holland OR hack de overheid OR (toegang! w/s (web OR internet)) OR qiy OR ioverheid OR iautoriteit OR (crisis! w/2 ICT!) OR (clearinghouse w/s botnet!) or (deltaplan w/s ict)	
Generic newspaper background corpus	Topic-related background corpus
rubricering 'classification'	a-film 'A-film'
internet 'Internet'	agendapunt 'item on agenda'
staden 'Staden'	westrozebeke 'Westrozebeke'
datum 'date'	ivm agendapunt 'concerning item on agenda'
google 'Google'	moorslede 'Moorslede'

An English translation is added for the topic titles and the suggested terms, for the reader's convenience. The queries have not been translated because they are only shown to illustrate which terms are already included

all experiments we have seen that FP and KLIP perform similarly. We already noted in Sect. 3.2 that the two methods are similar to each other. The a-symmetry of the KLIP function explains why its performance is a little better than FP in Fig. 1. This confirms the second part of our hypothesis.

We investigated the effect of the domain-specificity of the background corpus by comparing two background collections of different specificity for two tasks: For Personalized Query Suggestion we compared a domain-specific background collection of scientific literature (iSearch) to a background collection of general English language (COCA); and for QUINN we compared a topic-specific background collection to a more general background collection for the same genre (the Dutch-language newspaper collection Sonar). In the first case we found that the domain-specific background collection gave better results than the general-domain collection, and in the second case we found that the more general background collection gave convincingly better results than the highly specific corpus. This suggests that a background collection in the same language and genre as the foreground collection (such as English scientific articles or Dutch newspaper

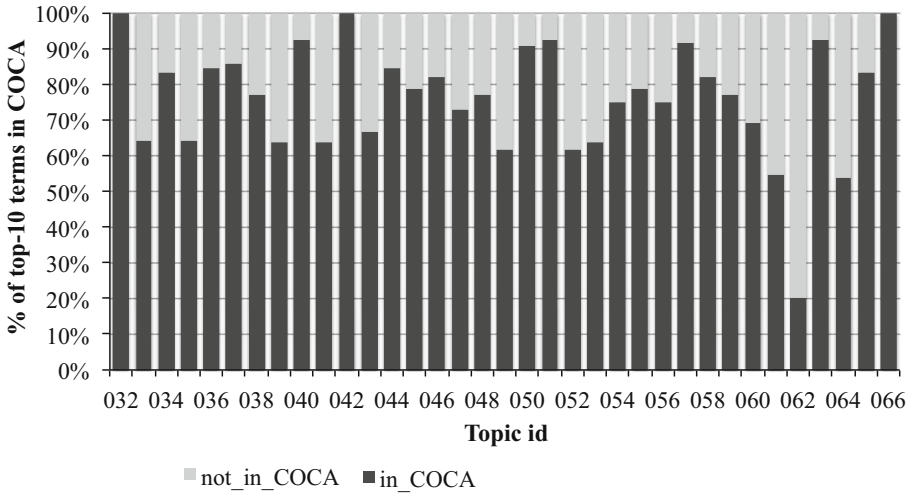

Fig. 7 The relative proportion of the top-10 terms generated by KLIP, FP and PLM for each of the topics in Personalized Query Suggestion that are present in the generic background corpus COCA. The coverage of the other background corpus, iSearch is not shown because it is 100 % in all cases

articles) gives good results, but a topic-specific background corpus seems a step too far in terms of domain-specificity.

In order to provide some more insight in the effect of the background collection on the generated terms, we analyzed the coverage of the background collections for the generated terms. The coverage of the background collection is relevant for term weighting because terms that do not occur in the background corpus are scored based on the frequency criterion only. In other words, the absence of a term in the background collection implies a high specificity of the term for the foreground collection. For relevant terms that are highly specific for a topic the absence in the background collection reflects their high specificity. However, if the coverage of the background corpus is too low, less relevant terms receive high scores because of their specificity relative to the background collection. We investigated the coverage of the background collections for the two tasks in this section.

In the case of Personalized Query Suggestion (Sect. 5.2.1) with iSearch as background corpus, all candidate terms are part of the background collection, since the foreground collection (set of retrieved documents) is a subset of the background collection. In the case of COCA as background corpus, not all candidate terms are part of the background collection, because COCA is an independent corpus. We compared the COCA word list with the top-ranked (top-10) terms that were generated by KLIP, FP and PLM for each of the topics. We found that on average, 76 % of the generated terms are present in COCA. Examples of these terms are *electron*, *mirror*, *cavity* and *pressure*. Examples of terms that are not in COCA are *ferromagnetic*, *waveguides*, *excitons* and *nanoclusters*. Figure 7 shows the proportion of the top-10 terms generated for the topics 032–065 that are present in the background collection COCA. For some topics, the terms are highly specific, e.g. topic 062 with terms such as *magnetohydrodynamic*, while for other topics the generated terms are much more frequent in general language, e.g. topic 042 with terms such as *electricity* and *energy*.

In the case of QUINN (Sect. 5.2.2), we first investigated the coverage of the SONAR corpus: We compared the SONAR word list with the top-ranked (top-10) terms that were

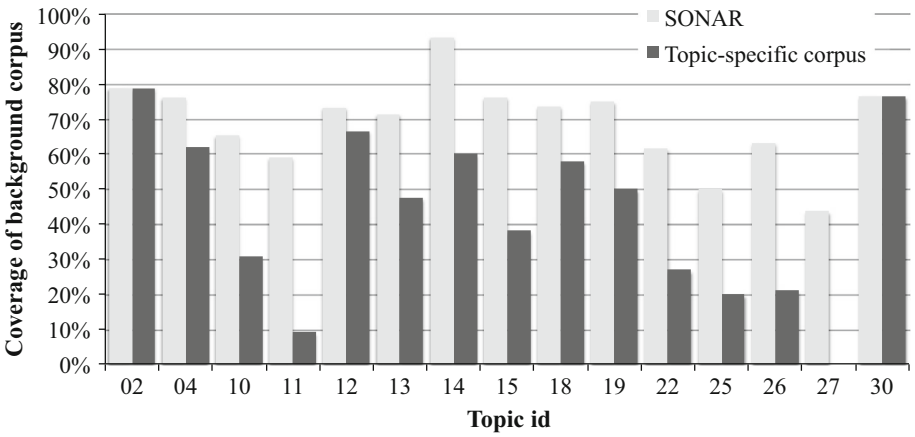

Fig. 8 The relative proportion of the top-10 terms generated by KLIP, FP and PLM for 15 topics in QUINN that are present in the external newspaper background corpus SONAR and the topic-specific background collection (older news articles for the same query)

Table 11 Summary of the coverage of the background corpora and their quality (for one example method, PLM)

Collection	Background	Coverage of top-10 terms (%)	Quality (PLM)
Personalized Query Suggestion	iSearch	100	**0.042**
	COCA	76	0.028
QUINN	SONAR	71	**11 %**
	Topic-specific	51	6 %

Note that the quality scores for Personalized Query Suggestion and QUINN cannot be compared to each other; they represent different measures. Boldface indicates the best scoring background corpus per collection

generated by KLIP, FP and PLM for each of the topics. We found that on average 71 % of the generated terms are present in SONAR. Examples of these terms are *rotterdam*, *studenten* ('students') and *lachgas* ('laughing gas'). The terms that are not in SONAR are very specific terms such as *schaliegas* ('schale gas'), spurious multi-word phrases such as *vooral kool* ('mainly cole'), and proper names such as *robin batens* and *tsipras*. Then we investigated the coverage of the topic-specific background collections (for each query a set of news articles retrieved for the same query but published earlier than the articles in the foreground collection). We found that on average, only 51 % of the terms are present in this specific background collection. Examples are again terms such as *rotterdam*, *studenten* ('students') en *ziekenhuis* ('hospital'). The terms that are not in the topic-specific background collection are in some cases again proper names such as *annelies* and spurious multiwords such as *greenpeace roept* ('greenpeace calls'), but also terms that are more general but did not occur in the small background collection for the topic, such as *zee* ('sea'), *wall street* en *goede doelen* ('charity funds'). Figure 8 shows a comparison between the coverage of both background corpora used for QUINN for 15 topics. It shows

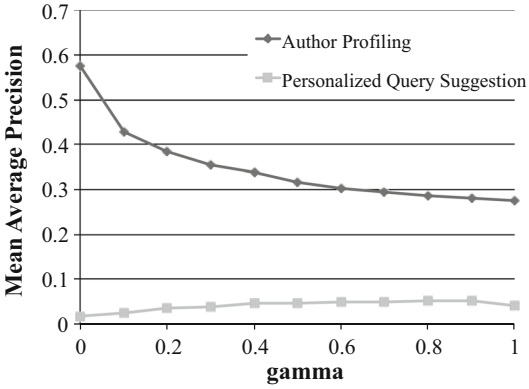

Fig. 9 The effect of the γ parameter in the KLIP method, regulating the balance between informativeness and phraseness in two collections: Author Profiling and Personalized Query Suggestion. The higher γ, the more weight the informativeness component has

Table 12 Example output of KLIP with different values of γ for one user in the Author Profiling collection and for one topic in the Personalized Query Suggestion collection

KLIP ($\gamma = 0.0$)	KLIP ($\gamma = 0.3$)	KLIP ($\gamma = 0.6$)	KLIP ($\gamma = 0.9$)
Author Profiling. Collection of scientific articles authored by one person, who has obtained a PhD in information retrieval. In a short CV, she describes her research topics as "entity ranking, searching in Wikipedia, and generating word/tag clouds."			
Entity ranking	Categories	Categories	Categories
Anchor text	Query	Query	Query
Relevance feedback	Documents	Documents	Documents
New york	Retrieval	Retrieval	Retrieval
Word clouds	Pages	Pages	Pages
Personalized Query Suggestion for one example topic (009). Information need: "I want information on how to measure dielectric properties on cells, for example in microfluidic systems."			
Biological cells	Dielectric	Dielectric	Dielectric
Alternating current	Biological cells	Cell	Cell
Elastomer actuators	Alternating current	Biological cells	Suspensions
Spectral representation	Elastomer actuators	Suspensions	Electrorheological
Low-frequency sub-dispersion depended	Cell	Electrorheological	Cells

a large diversion between topics, just as in the case of Personal Query Suggestion, but in all topics the SONAR corpus has a larger coverage than or equal to the topic-specific corpus.

A summary of the coverage of the background corpora and their quality (for one example method, PLM) is shown in Table 11. The table shows that for both tasks, the background collection with the highest coverage gives the best results.

5.3 What is the influence of multi-word phrases?

We investigate the balance between informativeness and phraseness for the two collections for which we have ground truth terms available: Author Profiling and Personalized Query Suggestion. We run KLIP on both collections. In Personalized Query

Suggestion, we use the iSearch collection as background corpus. We evaluate values for γ in Eq. 13 ranging from 0.0 (Phraseness only) to 1.0 (Informativeness only) with steps of 0.1 The results are in Fig. 9.

Again, we see that mean average precision is much lower for the Personalized Query Suggestion collection than for the Author Profiling collection. This is because the ground truth is very strictly defined in the Personalized Query Suggestion collection. More interestingly, the effect of gamma is very different between the two collections: the phraseness component should be given much more weight in the Author Profiling collection than in the Personalized Query Suggestion collection. This is surprising because the proportion of multi-word phrases in the ground truth set is very similar for both collections. We had a more detailed look at the output of KLIP for both collections to see what causes this difference. Table 12 shows the top-5 terms for one user in the Author Profiling collection and one topic in the Personalized Query Suggestion collection, ranked using KLIP with different values of γ.

The table shows that in the Author Profiling collection, multi-words have already disappeared from the top-5 when $\gamma = 0.3$, while in the Personalized Query Suggestion collection, three out of five terms are still multi-words for the same value of γ. Even if we set $\gamma = 1.0$ (informativeness only), the top-10 terms for the example topic still contains three multi-words.[16] A more detailed look of the output for both collections reveals that over all users and topics, more multi-words are extracted from the data in the Personalized Query Suggestion collection than in the Author Profiling collection (also using other term scoring methods). The most probable explanation for this is that each topic in the Personalized Query Suggestion data covers a very narrow domain. We extract terms from the documents that are relevant to this narrow domain. In these documents, some multi-word terms (e.g. 'biological cells') are highly frequent, not only compared to other multi-word terms but even compared to single-word terms.

5.3.1 Discussion: What is the influence of multi-word phrases?

In Sect. 5.1.1, we showed that the phraseness methods outperform the informativeness methods for author profiling. The reason is that in this collection, the human-defined ground truth has a large proportion of multi-word terms. The results confirm our hypothesis:

> **Hypothesis:** We expect C-Value and KLIP to give the best results for collections and use cases where multi-word terms are important. CB, PLM and FP are also capable of extracting multi-words but the scores of multi-words are expected to be lower than the scores of single-words for these methods. On the other hand, C-Value cannot extract single-word terms, which we expect to be a weakness because single-words can also be good terms.

When comparing informativeness methods and phraseness methods for a given collection, two aspects play a role: Multi-word terms are often considered to be better terms than single-word terms (see Sect. 5.3). On the other hand, multi-word terms have lower frequencies than single-word terms (see Sect. 3.1), which makes them sparse in small collections. In the case of a small collection, consisting of 1 or a few documents, the

[16] Recall that all n-grams with $n = \{1, 2, 3\}$ are candidate terms. This implies that multi-word terms can be selected based on the informativeness criterion only, even though their frequencies are relatively low compared to single-word terms.

frequency criterion will select mostly single-word terms. For that reason, KLIP performs better than C-Value. In addition to that, we also saw in Sect. 5.1.1 that KLP without the informativeness criterion also outperforms C-Value. As we pointed out in Sect. 3.3, both methods select terms on the basis of different criteria: In C-Value, the score for a term is discounted if the term is nested in frequent longer terms (e.g. the score for 'surgery clinic' would be discounted because it is embedded in the relatively frequent term 'plastic surgery clinic'). In KLP, on the other hand, the frequency of the term as a whole is compared to the frequencies of the unigrams that it contains; the intuition is that relatively frequent multi-word terms that are composed of relatively low-frequent unigrams (e.g. 'ad hoc', 'new york') are the strongest phrases. We found that the KLP criterion tends to generate better terms than the C-Value criterion.

In Sect. 5.3 we saw that if we combine informativeness and phraseness in one term scoring method, the optimal weight of the two components depends on the collection at hand. In general, the importance of multi-word phrases depends on three factors:

- Language. In compounding languages such as Dutch and German, noun compounds are written as a single word, e.g. *boottocht* 'boat trip'. In English, these compounds are written as separate words. As a result, the proportion of relevant terms that consist of multiple words is higher for English than for a compounding language such as Dutch. For example, the proportion of multi-words in the user-formulated Boolean queries for the Dutch collection QUINN is only 16 %. The proportions of multi-word phrases in the ground truth term lists for Author Profiling and Personalized Query Suggestion are 50 and 57 % respectively. This implies that (a) we cannot generalize the results in this paper to all languages and (b) although it is to be recommended to tune the γ parameter for any new collection, this is even more important in the case of a new language.
- Domain. In the scientific domain (in our case the Author Profiling and Personalized Query Suggestion collections), more than half of the user-selected terms are multi-word terms. A method with a phraseness component is therefore the best choice (KLIP with a low γ or C-Value) for collections of scientific English documents.
- Use case and evaluation method. For Author Profiling, multi-word terms are highly important if the profile is meant for human interpretation (such as keywords in a digital library, or on an author profile): human readers prefer multi-word terms because of their descriptiveness. This implies that when terms are meant for human interpretation, a method with a phraseness component is the best choice (KLIP with a low γ or C-Value). On the other hand, in cases where terms are used as query terms, single-word terms might be more effective, and PLM or FP would be preferable.

6 Conclusion

We investigated the influence of three factors in the success of a term scoring method in term extraction: collection size, background collection and the importance of multi-word terms. Below, we draw conclusions, remark the limitations of our study, and make recommendations below for each of the three factors.

With respect to the collection size, our results and analyses indicate that

- larger collections lead to better terms.

- for collections larger than 10,000 words, the best performing method for the task of author profiling is Kullback–Leibler divergence for Informativeness and Phraseness (KLIP) (Tomokiyo and Hurst 2003).
- for modeling smaller collections up to 5,000 words, the best performing method for the task of author profiling is Parsimonious Language Models (PLM) (Hiemstra et al. 2004). for PLM, we recommend to empirically choose (tune) the λ parameter for each new combination of foreground and background collections because the optimal value differs between collections and background corpora.

For collections smaller than 1,000 words we could not prove the success of any of the term scoring methods: all methods perform poorly on small corpus sizes, for both evaluation collections. We speculate that this is caused by the prominence of the frequency criterion in all methods: For small collections term frequency is a weak variable: most terms occur only once or a few times.

With respect to use of a background collection, our methodological analyses indicate that PLM would be a good choice in situations where the foreground collection or document is embedded in a larger collection, and KLIP would be a good choice for extracting terms from a larger collection that does not have an overarching background collection. However, we did not find strong evidence for one method being better than the others in all scenarios. For methods that require an external background collection, we recommend to use a collection with texts from the same language and genre as the foreground collection. We found for newspaper text that a generic newspaper collection would be preferable over a set of newspaper articles covering 1 particular topic. In further analyses, we found that for both evaluation collections, the background collection with the highest coverage of foreground terms gave the best results. In the case of Personalized Query Suggestion, one limitation of our work is that we worked with a strictly defined ground truth: a small set of human-formulated terms. This caused the evaluation scores for any background corpus to be low, and made it difficult to draw strong conclusions on the better choice of background corpus. More work is needed on finding the best strategy for Personalized Query Suggestion in a complex topic domain (Verberne et al. 2015a).

With respect to the importance of multi-word terms, our results and analyses indicate that KLIP is the most flexible method for extracting both single-word terms and multi-word terms. We introduced the parameter γ that weights the informativeness component relative to the phraseness component in KLIP and thereby determines the proportion of multi-word terms in the output. We recommend that the value of γ is empirically chosen per collection and goal. Overall, we have shown that extracting relevant terms using unsupervised term scoring methods is possible in diverse use cases, and that the methods are applicable in more contexts than their original design purpose. We especially obtained good results in the case of author profiling; automatically extracted terms could be used as suggestions to authors for creating an online profile or a summary for a digital library, in addition to manually formulated terms. The results obtained for automatic query expansion and query term suggestion were mixed, partly due to the small collection size and the domain-specific language use.

Our final recommendation is that the choice of method and evaluation for term extraction should depend on the specific use case. If there is a clearly defined goal, such as query expansion, then the evaluation measure for this goal can be exploited as extrinsic evaluation for the term scoring method. It should always be taken into account that the goal poses specific requirements on the extracted terms: terms that are informative for author profiling are different from terms that are powerful for query expansion. Thus, not only the

collection size, language and domain determine the success of a term scoring method, but also the context in which the terms are used – this context is not necessarily the purpose the method was designed for.

An interesting direction for future research would be to combine the strengths of multiple term scoring methods into one, flexible, method with tuneable parameters for the weight of the background collection (informativeness) and the importance of multi-word terms (phraseness).

Acknowledgments This publication was supported by the Dutch national program COMMIT (project P7 SWELL)

References

Azzopardi, L., Kelly, D., & Brennan, K. (2013). How query cost affects search behavior. In *Proceedings of the 36th international ACM SIGIR conference on Research and development in information retrieval*, ACM, pp 23–32.

Cao, G., Nie, J.Y., Gao, J., & Robertson, S. (2008). Selecting good expansion terms for pseudo-relevance feedback. In *Proceedings of the 31st annual international ACM SIGIR conference on research and development in information retrieval*, ACM, pp 243–250.

Carpineto, C., De Mori, R., Romano, G., & Bigi, B. (2001). An information-theoretic approach to automatic query expansion. *ACM Transactions on Information Systems (TOIS)*, *19*(1), 1–27.

Choi, S., & Choi, J. (2014). Exploring effective information retrieval technique for the medical web documents: Snumedinfo at clefehealth2014 task 3. In: *CLEF* (Working Notes), pp. 167–175.

Davies, M. (2009). The 385+ million word corpus of contemporary american english (1990–2008+): Design, architecture, and linguistic insights. *International Journal of Corpus Linguistics*, *14*(2), 159–190.

Debole, F., & Sebastiani, F. (2004). Supervised term weighting for automated text categorization. In S. Sirmakessis (Ed.), *Text mining and its applications* (pp. 81–97). Berlin: Springer.

Frantzi, K., Ananiadou, S., & Mima, H. (2000). Automatic recognition of multi-word terms: The c-value/nc-value method. *International Journal on Digital Libraries*, *3*(2), 115–130.

Goeuriot, L., Kelly, L., Li, W., Palotti, J., Zuccon, G., Hanbury, A., et al. (2014). SHARE/CLEF eHealth evaluation Lab 2014, task 3: User-centred health information retrieval. CLEF 2014 Online Working. *Notes*, *1180*, 43–61.

Gutwin, C., Paynter, G., Witten, I., Nevill-Manning, C., & Frank, E. (1999). Improving browsing in digital libraries with keyphrase indexes. *Decision Support Systems*, *27*(1), 81–104.

Hiemstra, D., Robertson, S., & Zaragoza, H. (2004). Parsimonious language models for information retrieval. In: *Proceedings of the 27th annual international ACM SIGIR conference on research and development in information retrieval*, ACM, pp 178–185.

Hofmann, K., Tsagkias, M., Meij, E., & De Rijke, M. (2009). The impact of document structure on keyphrase extraction. In: *Proceedings of the 18th ACM conference on Information and knowledge management,* ACM, pp 1725–1728.

Huang, C. K., Chien, L. F., & Oyang, Y. J. (2003). Relevant term suggestion in interactive web search based on contextual information in query session logs. *Journal of the American Society for Information Science and Technology*, *54*(7), 638–649.

Järvelin, K., & Kekäläinen, J. (2002). Cumulated gain-based evaluation of ir techniques. *ACM Transactions on Information Systems (TOIS)*, *20*(4), 422–446.

Java, A., Song, X., Finin, T., & Tseng, B. (2007). Why we twitter: Understanding microblogging usage and communities. In *Proceedings of the 9th WebKDD and 1st SNA-KDD 2007 workshop on Web mining and social network analysis*, ACM, pp 56–65.

Kelly, L., Goeuriot, L., Suominen, H., Schrek, T., Leroy, G., Mowery, D.L., et al. (2014). Overview of the share/clef ehealth evaluation lab 2014. In *Proceedings of CLEF 2014*, Springer, Lecture Notes in Computer Science (LNCS).

Kim, S. N., Medelyan, O., Kan, M. Y., & Baldwin, T. (2013). Automatic keyphrase extraction from scientific articles. *Language resources and evaluation*, *47*(3), 723–742.

Krauthammer, M., & Nenadic, G. (2004). Term identification in the biomedical literature. *Journal of biomedical informatics*, *37*(6), 512–526.

Luhn, H. P. (1957). A statistical approach to mechanized encoding and searching of literary information. *IBM Journal of research and development*, *1*(4), 309–317.

Lykke, M., Larsen, B., Lund, H., & Ingwersen, P. (2010). Developing a test collection for the evaluation of integrated search. In C. Gurrin, Y. He, G. Kazai, U. Kruschwitz, S. Little, T. Roelleke, S. Rüger, & K. van Rijsbergen (Eds.), *Advances in Information Retrieval* (Vol. 5993, pp. 627–630)., Lecture Notes in Computer Science Berlin/Heidelberg: Springer.

Matsuo, Y., & Ishizuka, M. (2004). Keyword extraction from a single document using word co-occurrence statistical information. *International Journal on Artificial Intelligence Tools*, *13*(01), 157–169.

Oh, H.S., & Jung, Y. (2014). A multiple-stage approach to re-ranking clinical documents. In *CLEF* (Working Notes), pp 210–219.

Oostdijk, N., Reynaert, M., Monachesi, P., Van Noord, G., Ordelman, R., Schuurman, I., et al. (2008). From d-coi to sonar: a reference corpus for dutch. In LREC.

Ortega, J. L., & Aguillo, I. F. (2014). Microsoft academic search and google scholar citations: Comparative analysis of author profiles. *Journal of the Association for Information Science and Technology*, *65*(6), 1149–1156.

Rayson, P., & Garside, R. (2000). Comparing corpora using frequency profiling. In *Proceedings of the workshop on Comparing Corpora*, Association for Computational Linguistics, pp. 1–6.

Salton, G. (1968). Automatic information organization and retrieval. New York: McGraw Hill.

Salton, G., & Buckley, C. (1988). Term-weighting approaches in automatic text retrieval. *Information Processing & Management*, *24*(5), 513–523.

Salton, G., Wong, A., & Yu, C. T. (1976). Automatic indexing using term discrimination and term precision measurements. *Information Processing & Management*, *12*(1), 43–51.

Shen, W., Nie, J.Y., & Liu, X. (2014). An investigation of the effectiveness of concept-based approach in medical information retrieval grium@ clef2014ehealthtask 3. *Proceedings of the ShARe/CLEF eHealth Evaluation Lab*.

Shen, X., Tan, B., & Zhai, C. (2005). Implicit user modeling for personalized search. In *Proceedings of the 14th ACM international conference on Information and knowledge management*, ACM, pp 824–831.

Sparck, J. K. (1972). A statistical interpretation of term specificity and its application in retrieval. *Journal of documentation*, *28*(1), 11–21.

Tomokiyo, T., Hurst, M. (2003). A language model approach to keyphrase extraction. In *Proceedings of the ACL 2003 workshop on Multiword expressions: analysis, acquisition and treatment* (Vol. 18, pp. 33–40) Association for Computational Linguistics.

Trieschnigg, D., Pezik, P., Lee, V., De Jong, F., Kraaij, W., & Rebholz-Schuhmann, D. (2009). Mesh up: effective mesh text classification for improved document retrieval. *Bioinformatics*, *25*(11), 1412–1418.

Verberne, S. (2014). A language-modelling approach to user-centred health information retrieval. In *Proceedings of the ShARe/CLEF eHealth evaluation Lab* (CLEF2014 Working Notes), pp 269–275.

Verberne, S., Sappelli, M., & Kraaij, W. (2013). Term extraction for user profiling: Evaluation by the user. In *Late-breaking results, project papers and workshop proceedings of the 21st conference on user modeling, adaptation, and personalization* (UMAP).

Verberne, S., Sappelli, M., & Kraaij, W. (2014). Query term suggestion in academic search. In *Advances in information retrieval. 36th European conference on IR research, ECIR 2014, Amsterdam, The Netherlands* (Vol. 8416, pp. 560–566), April 13–16, 2014. Proceedings. Berlin: Springer.

Verberne, S., Sappelli, M., Järvelin, K., & Kraaij, W. (2015a). User simulations for interactive search: Evaluating personalized query suggestion. In *Advances in Information Retrieval. 37th European Conference on IR Research, ECIR 2015* (Vol. 9022, pp. 678–690) Vienna, Austria, March 29–April 2, 2015. Proceedings.

Verberne, S., Wabeke, T., & Kaptein, R. (2015b). Quinn: Query updates for news monitoring. In *Proceedings of the 14th Dutch-Belgian Information Retrieval Workshop (demo paper)*, p 30.

Witten, I.H., Paynter, G.W., Frank, E., Gutwin, C., & Nevill-Manning, C.G. (1999). Kea: Practical automatic keyphrase extraction. In *Proceedings of the Fourth ACM Conference on Digital Libraries*, ACM, New York, DL '99, pp. 254–255, doi:10.1145/313238.313437.

Xu, J., & Croft, W.B. (1996). Query expansion using local and global document analysis. In *Proceedings of the 19th annual international ACM SIGIR conference on Research and development in information retrieval*, ACM, pp 4–11.

Zhai, C. (2008). Statistical language models for information retrieval. *Synthesis Lectures on Human Language Technologies*, *1*(1), 1–141.

Zhu, M. (2004). *Recall, precision and average precision*. Department of Statistics and Actuarial Science, University of Waterloo, Waterloo Working paper.